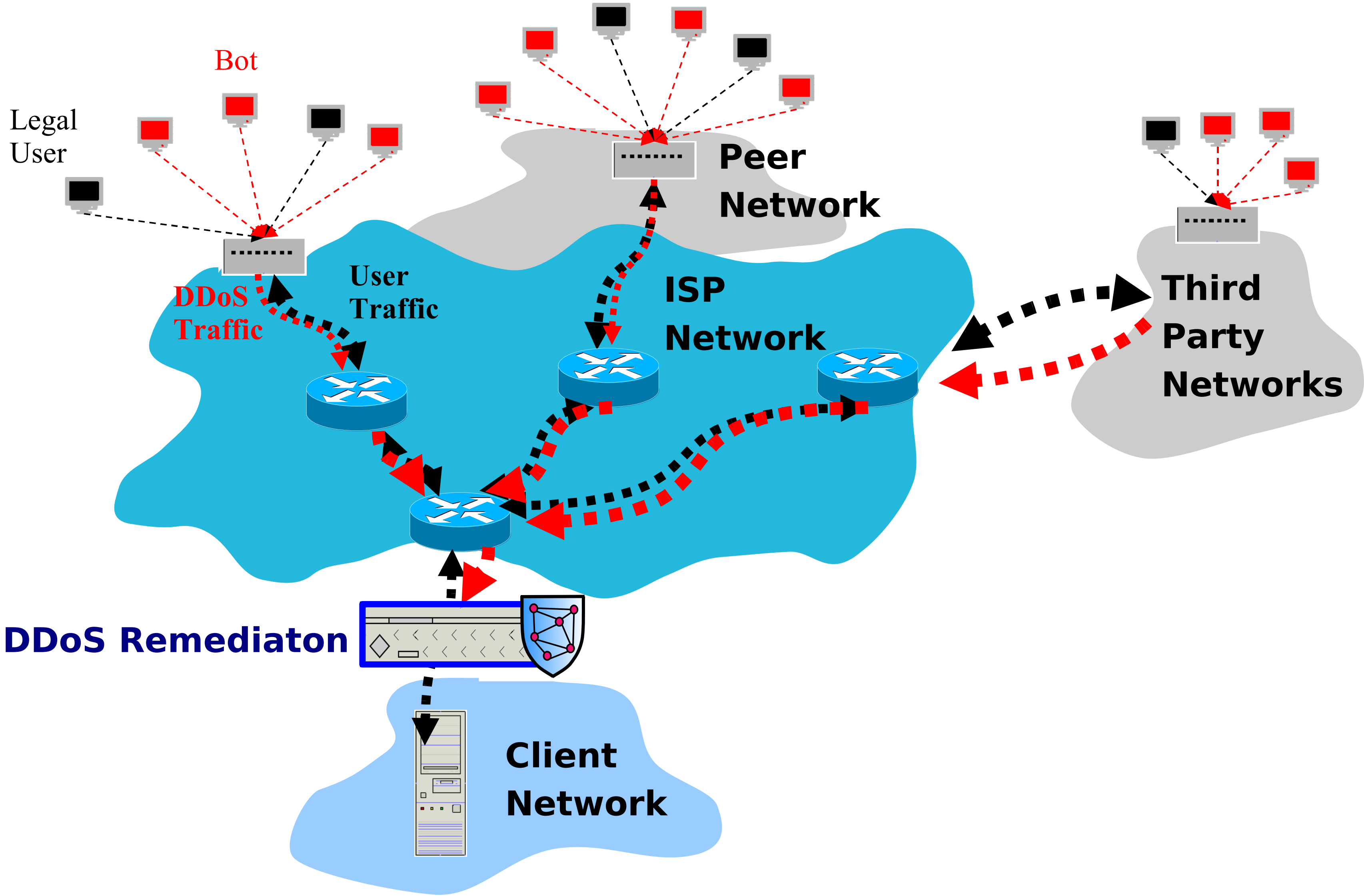
Bayes Optimal DDoS Mitigation by Adaptive History-Based IP Filtering

Markus Goldstein,
Christoph Lampert, Matthias Reif, Armin Stahl and Thomas Breuel

German Research Center for Artificial Intelligence (DFKI)
Image Understanding and Pattern Recognition Group
Kaiserslautern, Germany

IUPR

Near-Target DDoS Attack Mitigation



Overview

1. Current DDoS Remediaton Approaches and Solutions
2. Bayes Optimal Packet Filtering
3. Adaptive Attack Adjustment
4. Experimental Evaluation
5. Conclusion and Future Work

DDoS Mitigation
Existing Approaches

- Ingress Filtering (RFC 2827)
  - Near-source solution
  - Protects against IP spoofing
- Infrastructure based Approaches
  - Requires modified routers for packet marking
  - Savage et al: IP Traceback (SIGCOMM 2000)
  - Protect against IP spoofing
- History-based IP Filtering (Peng et al., ICC 2003)
  - Build IP address database during regular operation mode
  - Deny all “new” addresses during DDoS attacks
- Source Address Prefix Clustering (Pack et al., SecureComm 2006)
  - First IP density estimation approach

DDoS Mitigation
Existing Approaches

- Outlier Detection
  - Outlier detection: PCA, Clustering, Bagging, Active Learning
  - Used by commercial systems like Radware, Cisco, Arbor
  - Requires protocol understanding and many, many rules
- Attack Detection
  - Required by all approaches to enable remediation mode
  - Not focus of this work
  - Many approaches
    - Packet/ flow rate counting
    - Change-point detection (i.e. **CUSUM**)
    - Wavelet analysis
    - Statistical methods: PCA, Clustering

Bayes Decision Theory

- Idea: History-based IP filtering, but use **probability estimations** for legality $P(x|L)$ of a source IP address
- Minimize Bayes risk for decision function α:

$$\text{risk}(\alpha) = \sum_{x \in X} \text{loss}(\alpha(x)|x) P(x).$$

with using the loss matrix λ

$\lambda(\omega\|y)$	legal	illegal
accept	0	ϵ
reject	1	0

leading to an optimal packet classifier

$$\text{risk}(\alpha) = P(\mathtt{L}) \sum_{\{\alpha(x)=\mathtt{R}\}} P(x|\mathtt{L}).$$

Bayes Decision Theory

- Idea: History-based IP filtering, but use **probability estimations** for legality $P(x|L)$ of a source IP address
- Minimize Bayes risk for decision function α:

$$\mathrm{risk}(\alpha) = \sum_{x \in X} \mathrm{loss}(\alpha(x)|x) P(x).$$

Expected loss sums up all costs times their probability

with using the loss matrix λ

$\lambda(\omega\|y)$	legal	illegal
accept	0	ϵ
reject	1	0

leading to an optimal packet classifier

$$\mathrm{risk}(\alpha) = P(\mathtt{L}) \sum_{\{\alpha(x)=\mathtt{R}\}} P(x|\mathtt{L}).$$

Bayes Decision Theory

- Idea: History-based IP filtering, but use **probability estimations** for legality $P(x|L)$ of a source IP address
- Minimize Bayes risk for decision function α:

$$\mathrm{risk}(\alpha) = \sum_{x \in X} \mathrm{loss}(\alpha(x)|x)P(x).$$

with using the loss matrix λ

$\lambda(\omega\|y)$	`legal`	`illegal`
`accept`	0	ϵ
`reject`	1	0

$\varepsilon=0$ under the assumption that target is not overloaded

leading to an optimal packet classifier

$$\mathrm{risk}(\alpha) = P(\mathtt{L}) \sum_{\{\alpha(x)=\mathtt{R}\}} P(x|\mathtt{L}).$$

Practical Filtering

- To minimize risk, we drop requests with the lowest $P(x|L)$
- Since risk only increases while dropping requests, we let N requests pass (as much as the server could handle):

$$\alpha^*(x_i) := \begin{cases} \texttt{reject} & \text{if } x_i \text{ is one of the } M-N \\ & \text{requests with lowest } P(x_i|\mathtt{L}), \\ \texttt{accept} & \text{otherwise,} \end{cases}$$

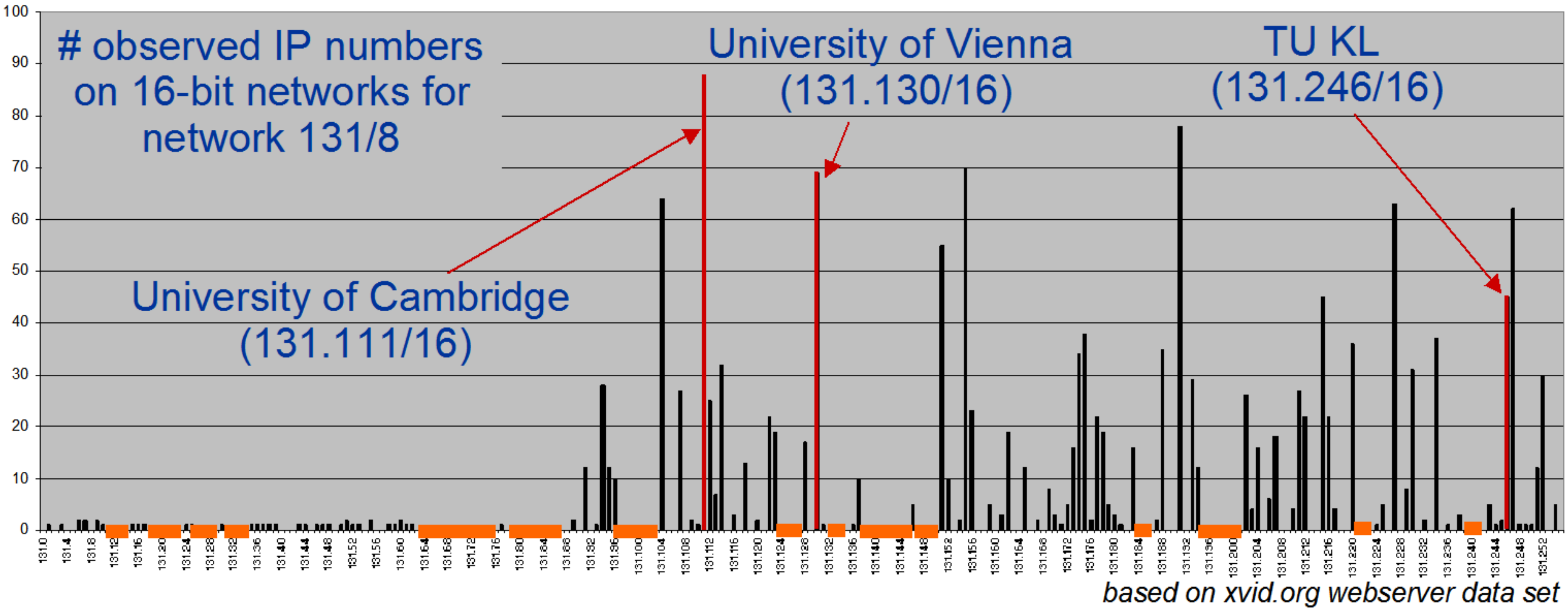
- For practical filtering, we define a probability threshold θ

$$\alpha_\theta(x) := \begin{cases} \texttt{accept} & \text{if } P(x|\mathtt{L}) \geq \theta, \\ \texttt{reject} & \text{if } P(x|\mathtt{L}) < \theta, \end{cases}$$

- $P(x|L)$ is estimated **in our case** from histograms of 16-24bit networks using historical traffic

Adaptivity

- The probability threshold θ is adjusted to the attack strength and the server capacity.



observed IP numbers on 16-bit networks for network 131/8

University of Cambridge (131.111/16)

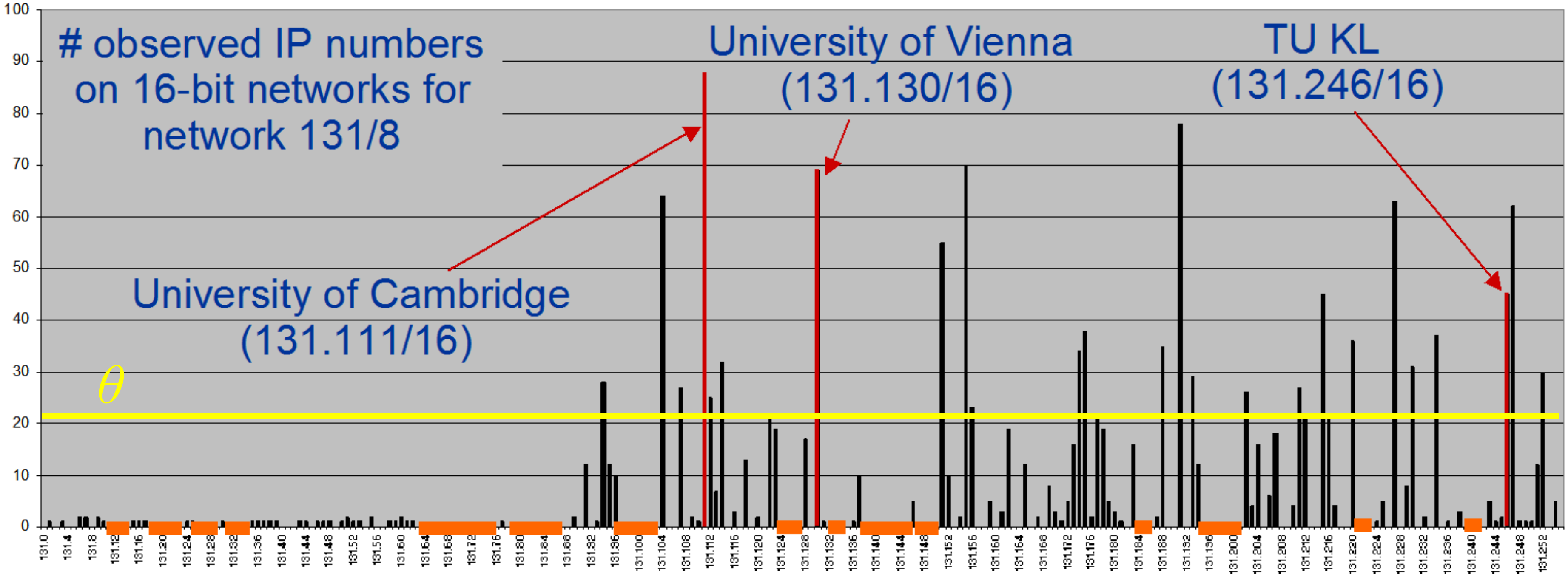
University of Vienna (131.130/16)

TU KL (131.246/16)

based on xvid.org webserver data set

Adaptivity

- The probability threshold θ is adjusted to the attack strength and the server capacity.



observed IP numbers on 16-bit networks for network 131/8

University of Vienna (131.130/16)

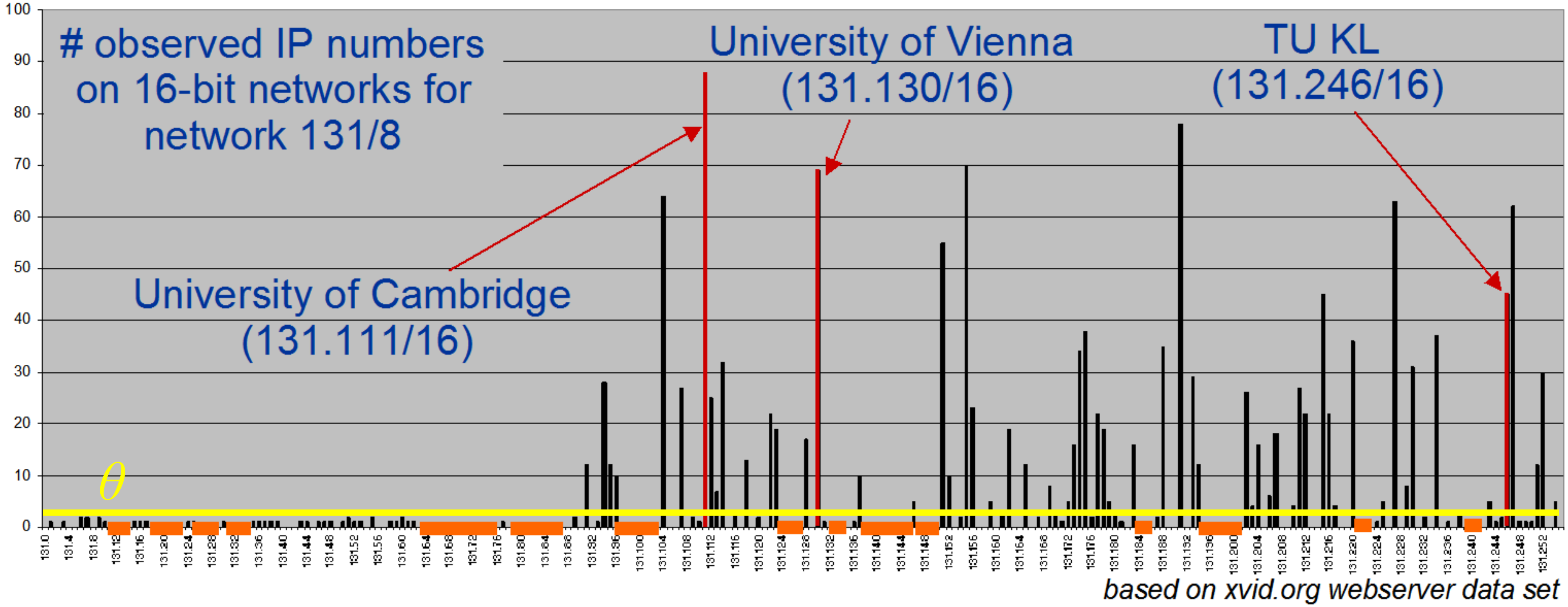
TU KL (131.246/16)

University of Cambridge (131.111/16)

based on xvid.org webserver data set

Adaptivity

- The probability threshold θ is adjusted to the attack strength and the server capacity.



observed IP numbers on 16-bit networks for network 131/8

University of Cambridge (131.111/16)

University of Vienna (131.130/16)

TU KL (131.246/16)

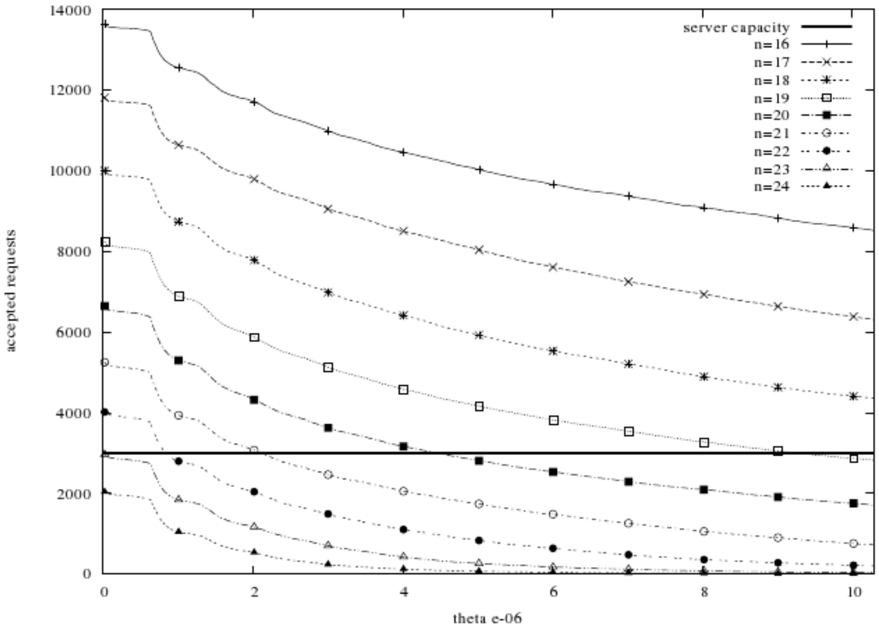
based on xvid.org webserver data set

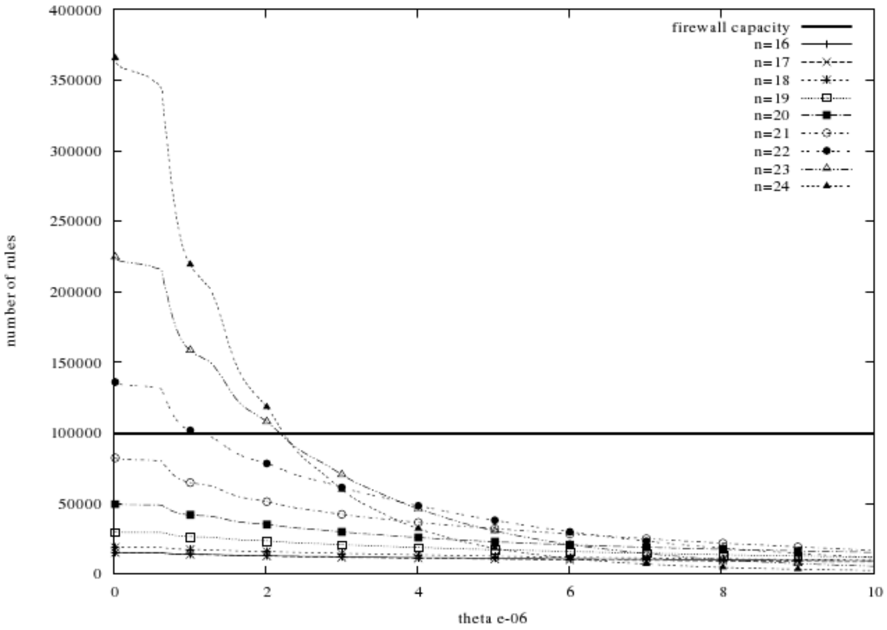
Data Sets

- DS-1 from www.xvid.org
  - 100 days of HTTP logfiles
  - 54 million requests from 1.3 million different IPs
  - Assume server could handle 3,000 rps
- DS-2 from mid-sized international web-community
  - 100 days of tcpdump data
  - 8 million requests from 145,000 different IPs
  - Assume server could handle 1,000 rps
- Artificially generated DDoS attack
  - 10 days lasting (-> 90 days of training left)
  - Bot network comprising of 100,000 attackers with a total capacity of 40,000 rps.
- Assume firewall restriction of 100,000 rules maximum

Results

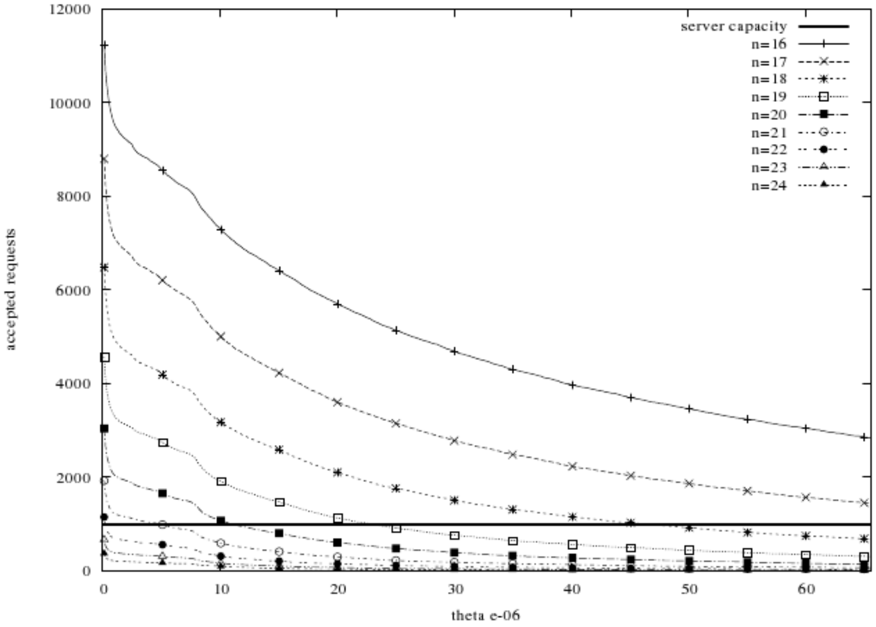
- Results for DS-1 (www.xvid.org)
- Use 21 bit networks to fulfill firewall restriction

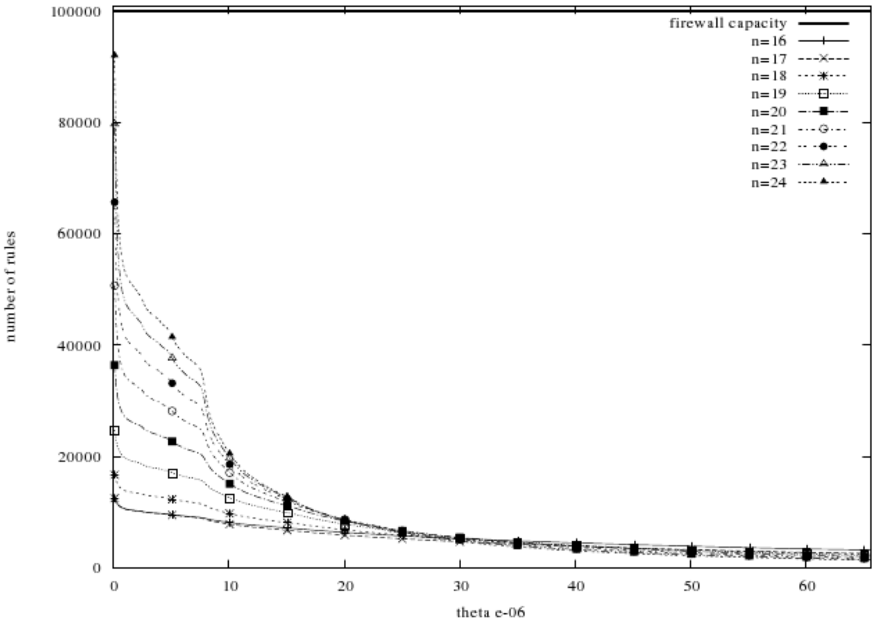




Results

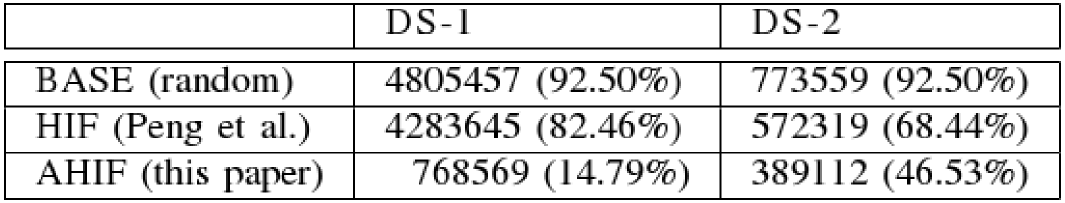
- Results for DS-2 (web-community)
- Firewall restriction does not apply, pick 23 or 24 bit network mask





Results: Collateral Damage

- Results with respect to *collateral damage*



	DS-1	DS-2
BASE (random)	4805457 (92.50%)	773559 (92.50%)
HIF (Peng et al.)	4283645 (82.46%)	572319 (68.44%)
AHIF (this paper)	768569 (14.79%)	389112 (46.53%)

Conclusion

- Advantages of our proposed method
  - Minimizes *collateral damage*
  - Adjusts to changing attack strength and sources
  - Can be applied with spoofed and highly distributed attacks
  - General statistically founded framework
  - Firewall rules can be prepared (periodically) **before** a DDoS attack is going on
- Extensions
  - Using IP density estimation for a better estimation of $P(x|L)$
  - Using multiple other features for estimating $P(x|L)$, i.e. country information, rates or URL information (i.e. with a Bayesian network)
  - Implementation of a Linux Kernel module for using an almost unrestricted amount of rules (*will be released as Open Source soon*)

Conclusion

Thanks for your attention! Questions?

www.iupr.org
goldstein@iupr.dfki.de